SECURITY TECHNOLOGY BASIC COURSE

BASIC AND APPLICATIONS OF SECURITY TECHNOLOGIES FOR EMBEDDED SYSTEMS

DATE: DECEMBER 14, 2023
NAME EPSG/SRCTE/SSCC TAKASHI KITAGAWA
RENESAS ELECTRONICS CORPORATION

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

RENESAS

CONTENT

No Voice
for this page

© 2023 Renesas Electronics Corporation. All rights reserved.

AIMS AND GOALS OF THIS LECTURE

No Voice
for this page

Aims

Significant changes in the connectivity environment represented by IoT (Internet of Things) have greatly improved the convenience of embedded systems as well as PCs and smartphones. On the other hand, the installation of a security function is essential for safe and secure use of networked equipment.

In this lecture, lectures are given on basic technologies for developing safe embedded systems from basic technologies such as cryptography and authentication algorithms to various attack cases and countermeasures, as well as security authentication schemes.

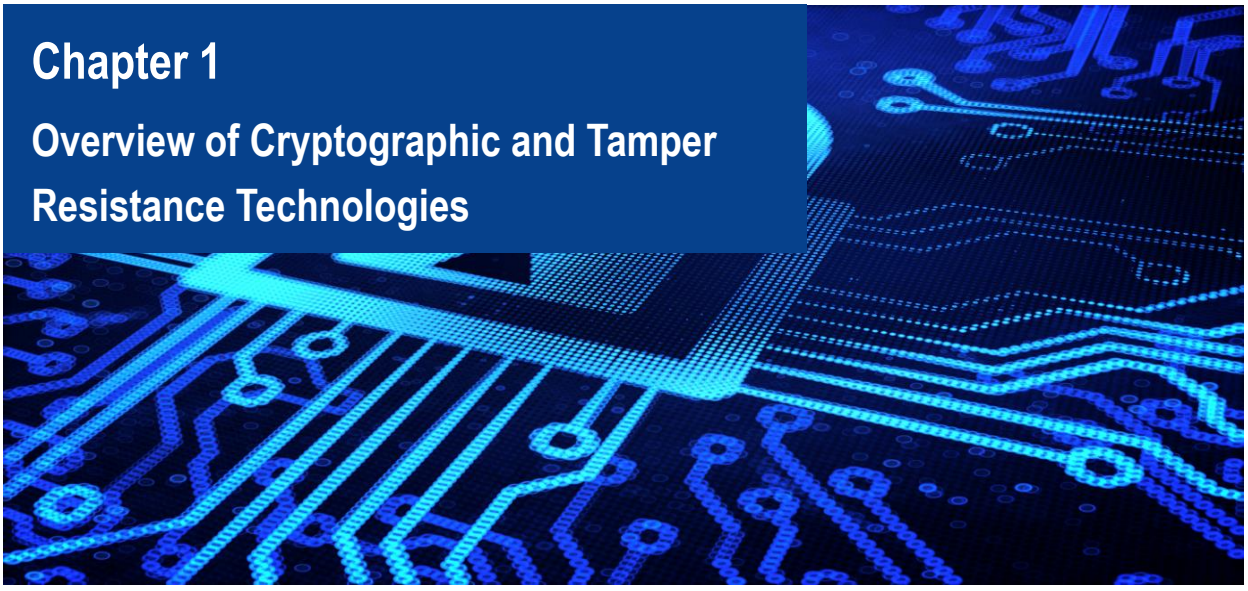
Goals

When applying security technology to a product, it becomes possible to conduct a basic consideration by referring to the latest information.

© 2023 Renesas Electronics Corporation. All rights reserved.

Chapter 1

Overview of Cryptographic and Tamper Resistance Technologies



© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

OVERVIEW OF THIS CHAPTER

This chapter explains fundamental technologies required for embedded security (mainly cryptographic technologies).

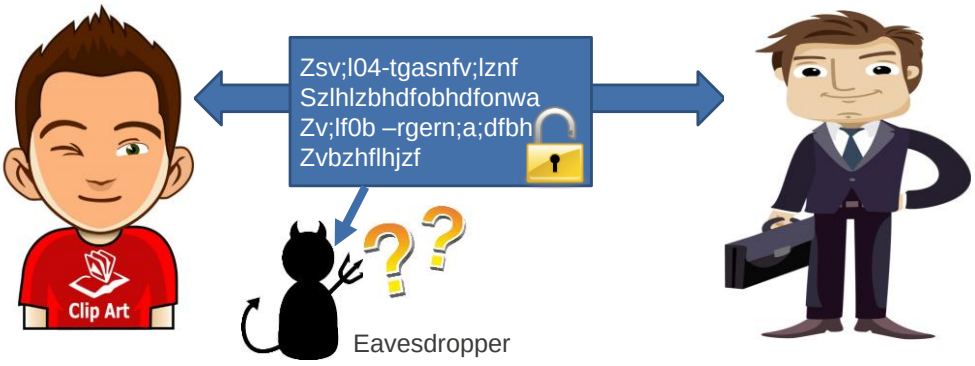
- Cryptographic primitive
  - Symmetric key and Public key cryptography, Hybrid encryption
  - Key exchange protocol
  - Cryptographic hash function
  - Messages authentication code (MAC) and Digital signature
  - Random number generator
- Authentication protocol
- Attack methods against cryptography and its computational complexity.
- Physical attacks against cryptographic LSIs
- Post-Quantum Cryptography (PQC)

© 2023 Renesas Electronics Corporation. All rights reserved.

This chapter explains the cryptographic technologies required for embedded security. First, we explain various cryptographic primitives. Cryptographic primitives are the basic component for implementing various cryptographies. Next, this chapter explain an authentication protocol, attack method for the authentication protocol, and physical attacks against the hardware-implemented cryptography. Finally, we explain Post-Quantum Cryptography as the latest topic.

WHAT IS CRYPTOGRAPHY?

- A technology that transforms messages so that they are not eavesdropped on communication channels, etc.



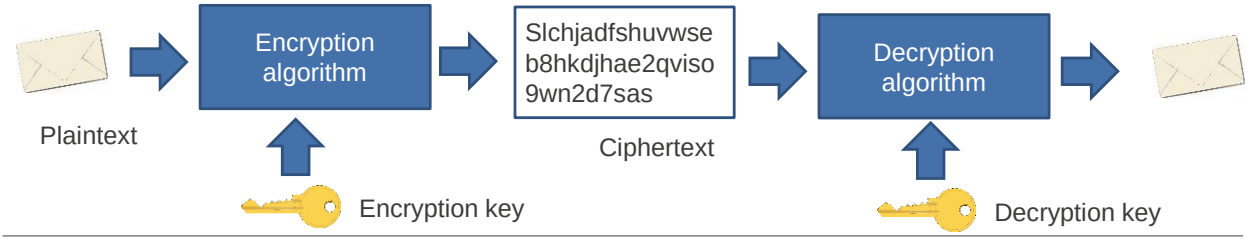
- Technologies such as digital signature, message authentication code, and cryptographic hash function are broadly referred to as cryptographic technologies.

© 2023 Renesas Electronics Corporation. All rights reserved.

In a narrow sense, encryption means technology that transforms messages so that third parties cannot read the contents even if the third parties intercepts the communication. In addition, related technologies such as digital signature, message authentication codes, and cryptographic hash function may also be included in cryptography in a broad sense. These will be described later.

CRYPTOGRAPHIC TERM

- Plaintext: The original document before encryption or the document after decryption the ciphertext.
- Ciphertext: The encrypted document.
- Encryption: Transform the plaintext to the ciphertext.
- Decryption: Transform the ciphertext to the plaintext.
- Encryption key: A key used for encryption.
- Decryption key: A key used for decryption (the encryption key differs from the decryption key for public key encryption)



© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

This page summarizes cryptographic terms. In cryptography, the original text before encryption is called plaintext. The conversion of plaintext into ciphertext is called encryption, and the conversion of ciphertext into plaintext is called decryption. An encryption key is the key used for encryption. Decryption key is the key used for decryption. A symmetric key cryptography uses the same key for encryption and decryption. A public key cryptography uses different keys for encryption and decryption.

KERCKHOFFS'S PRINCIPLE

- When a key and the decryption algorithm are revealed to an attacker, the ciphertexts are decrypted by attacker.
  - Keys must be shared secretly among the parties.
  - Should the algorithm be hidden?

Kerckhoffs's principle

Principle in cryptographic design proposed by the Dutch cryptographer Kerkhoff in the nineteenth century

> *The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.*

Cryptography must be secure even if the algorithm is exposed and known to an attacker.

© 2023 Renesas Electronics Corporation. All rights reserved.

A nineteenth-century Dutch cryptographer, Kerkhoff said, "Even if the algorithms are known to attackers, cryptography should be a safe method unless key is known." In fact, many modern cryptographic algorithms are made public.

REASONS WHY ALGORITHMS ARE OPEN IN THE MODERN CRYPTOGRAPHY

■ There are two reasons: Usability and Security.

- Cryptographic communications are necessary in our society. If cryptographic algorithm is not made public, we cannot use cryptographic communications across manufactures and countries.

Usability

- Publication of the algorithm allows security researchers to verify its security.
- Cryptography whose security is based on the secrecy of the algorithm has the risk of completely losing its security if the algorithm is exposed.

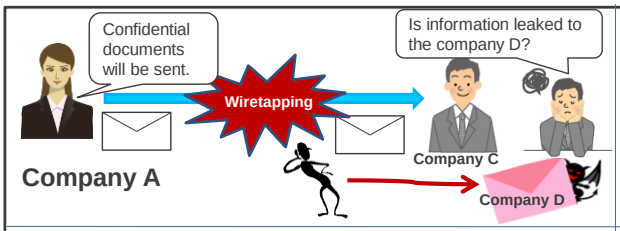
Security

Security of cryptographic algorithms should not be relied on secrecy of the algorithm.

© 2023 Renesas Electronics Corporation. All rights reserved.

The reason why modern cryptographic algorithms are made public is that sharing the same cryptographic algorithms globally is needed to conduct encrypted communications with a wide variety of parties. Also, the internationally formalized algorithm is also more reliable than the private algorithm because it passes the security verification by many experts. As a reminder, when the published encryption algorithms are implemented into a product, it is not necessary to disclose the implementation details.

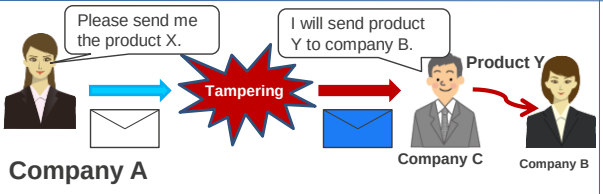
THREE ELEMENTS OF INFORMATION SECURITY



Protection against eavesdropping → Cryptography

Confidentiality

Ensure that confidential information is not disclosed.



Protection against tampering → Digital signature or message authentication code

Integrity

Make it possible to detect data tampering.

Keeps reliability of the system→ Duplication, redundancy

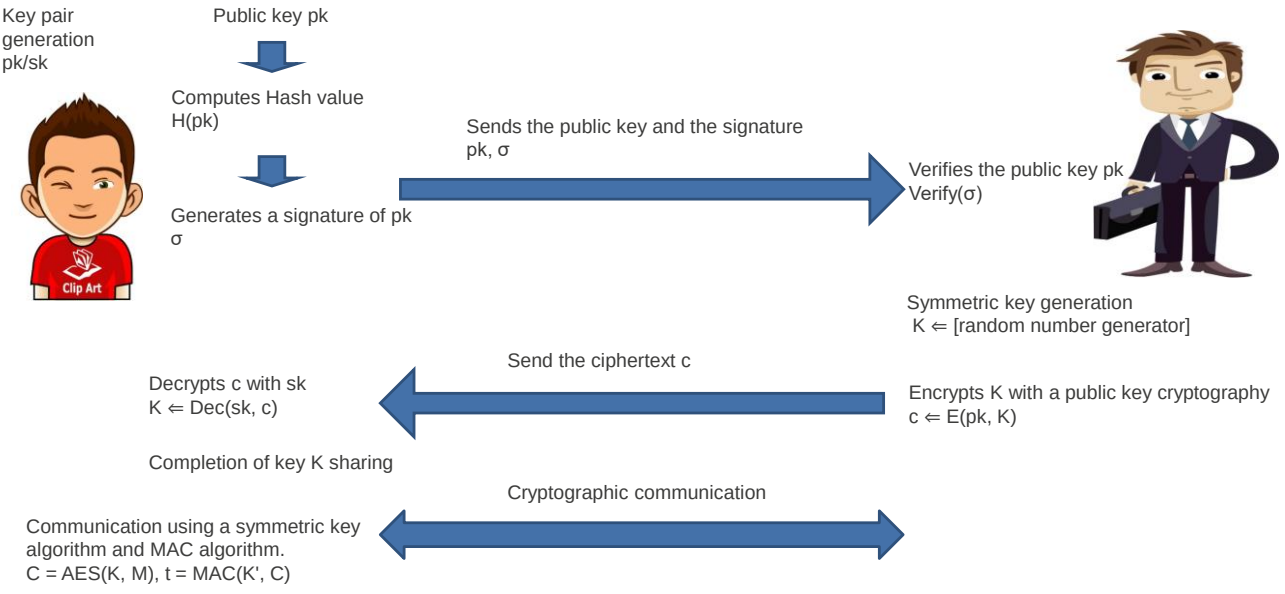
Availability

Users can consistently and readily access the resources.

© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

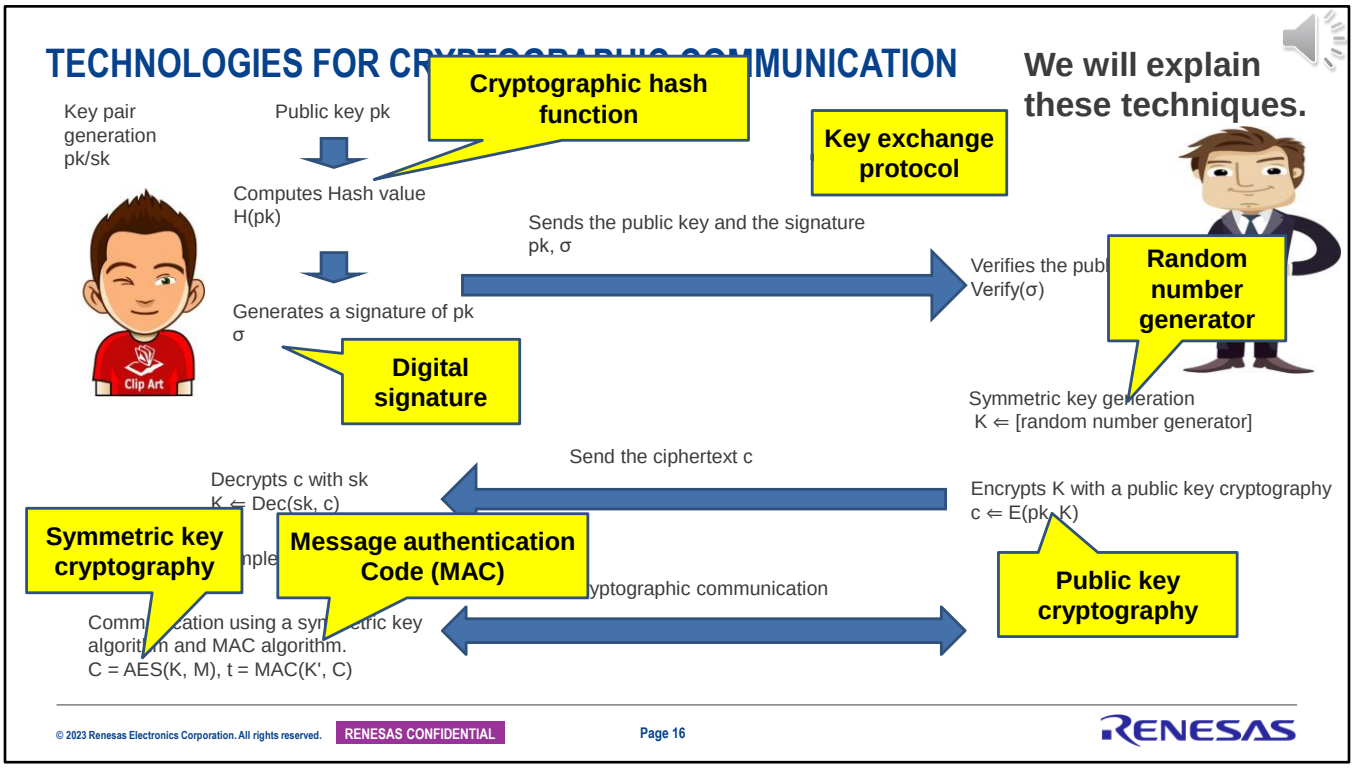
There are three important factors in discussing information security. Confidentiality means that confidential information is not disclosed to any third party. Measures such as encrypting communication data are taken to satisfy confidentiality. Integrity means that it is possible to detect tamper. Measures such as adding a digital signature to data are taken to ensure integrity. Availability means that access to information and functionality is unobstructed. Measures such as multiplexing the system are taken to ensure availability. The various cryptographic primitives described on the following pages are used to ensure confidentiality and integrity in the three important factors.

TECHNOLOGIES FOR CRYPTOGRAPHIC COMMUNICATION



© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

This page shows how various cryptographic technologies are used in cryptographic communication.



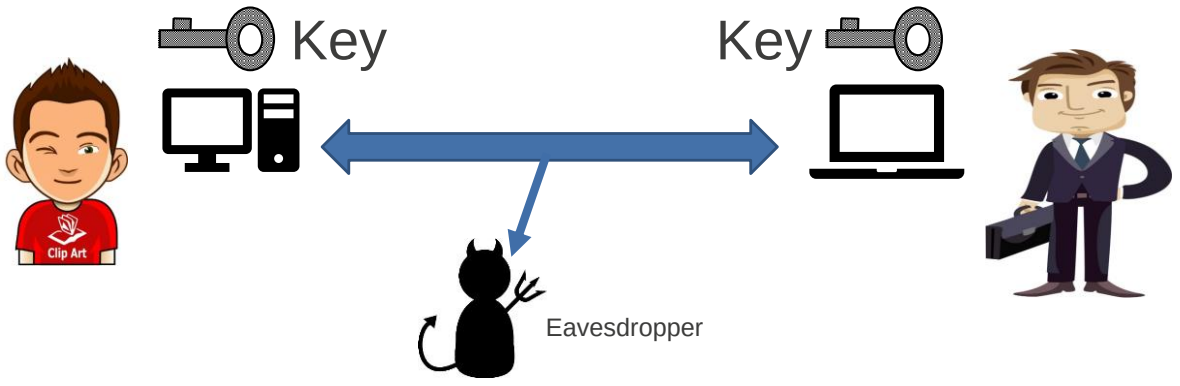
As you can see, cryptographic primitives: cryptographic hash function, digital signature, key exchange protocol, random number generation, public key cryptography, symmetric key cryptography, and message authentication codes are necessary for building secure communication channel. Next, we explain the detail of each cryptographic primitive.

SYMMETRIC KEY CRYPTOGRAPHY

© 2023 Renesas Electronics Corporation. All rights reserved.

SYMMETRIC KEY CRYPTOGRAPHY

Symmetric key cryptography uses the same key for encryption and decryption



- A key must be shared securely between parties in advance.
  - Sharing a key is conducted by a public key cryptography or a key exchange protocol (Diffie-Hellman).

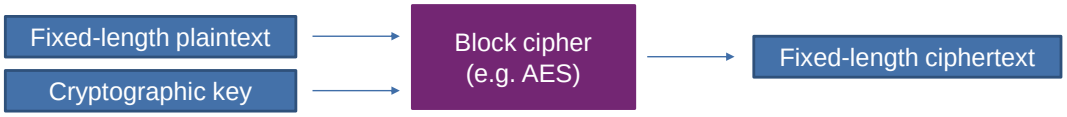
© 2023 Renesas Electronics Corporation. All rights reserved.

Symmetric key cryptography is a cryptographic algorithm where an encryption key and a decryption key are the same. To use symmetric key cryptography, a key must be shared safely between related parties without leaking the key to other parties. However, it is difficult to share the key to a related party secretly when conducting cryptographic communications with an unspecified number of people over the Internet. As methods to solve this problem, Diffie-Hellman key exchange algorithm, public key cryptography, and so on exist. The Diffie-Hellman key exchange algorithm and the public key cryptography are explained later.

AES (ADVANCED ENCRYPTION STANDARD)

- AES is a symmetric key cryptography standardized by the National Institute of Standards and Technology (NIST) as FIPS-197 in 2001.
  - ✓ The predecessor DES (Data Encryption Standard) became insecure due to its short key length of 54 bits. And NIST issued a public call for a new symmetric key encryption.
  - ✓ Rijndael was selected from 21 applications submitted from around the world.
- Characteristics
  - 128-bit block cipher (Block cipher encrypts a message by dividing it into defined lengths)
  - 128, 192, and 256 bits key lengths are available. The algorithms slightly differ depending on the key length.

Encryption of a block cipher



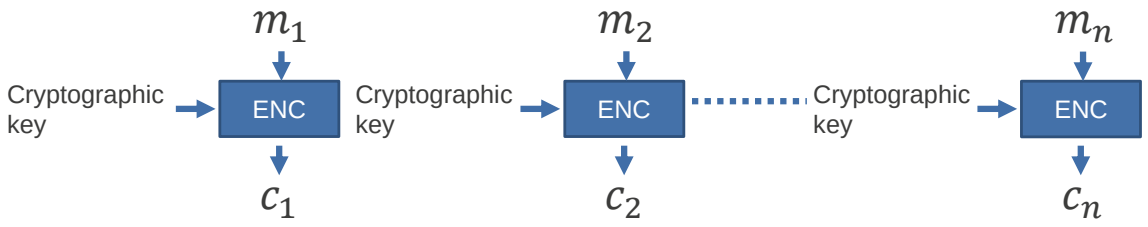
© 2023 Renesas Electronics Corporation. All rights reserved.

The most used symmetric key cryptography algorithm is AES. AES was adopted in 2001 by the U.S. government agency NIST through a public offering. This algorithm is standardized in FIPS197. One of the algorithm's characteristics is a 128-bits block cipher. The block cipher is the method in which messages are separated into blocks and encrypted for each block. AES encrypts the messages every 128 bits. Another characteristic is that the key length can be selected from 128, 192, or 256 bits.

BLOCK CIPHER MODES OF OPERATION

- To encrypt a longer plaintext using a block cipher, it is **not** recommended to apply the block cipher to each plaintext block as follows:
  - Divides a plaintext m into $m = m_1||m_2||\cdots||m_n$.
  - The ciphertext $c = ENC(m_1)||ENC(m_2)||\cdots||ENC(m_n)$.

This block cipher mode is called ECB mode.



© 2023 Renesas Electronics Corporation. All rights reserved.

A block length of AES is 128 bits, but there is a case that uses encrypts messages longer than the block length. To encrypt such messages, block cipher mode of operation is specified. The simplest cipher mode of operation is a method in which the messages are separated into blocks and encrypted for each block. This method is called ECB mode.

PROBLEM WITH ECB MODE

Original image

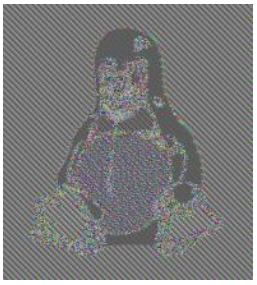


Image encrypted in the ECB mode

Taken from Wikipedia:
Block cipher mode of operation

Although the image is encrypted, the original image can be seen.
⇒ The plaintext is disclosed from the ciphertext.

© 2023 Renesas Electronics Corporation. All rights reserved.

However, it is not recommendable to use ECB mode to encrypt messages. This page shows an example of encryption of a penguin image using ECB mode of AES. You can see that the outline of the original image remains in the encrypted image. In other words, it means plaintext information leaks out from the ciphertext.

PROBLEM WITH ECB MODE

- Problem with the ECB mode
  - In the ECB mode, each plaintext block is encrypted separately.
  - The same ciphertexts are computed from blocks with the same value.
  - As you can see the example in the previous slide, the background of the image is white, thus the ciphertexts of the background has the same value.

In the ECB-mode, ciphertexts of the same plaintexts are the same.

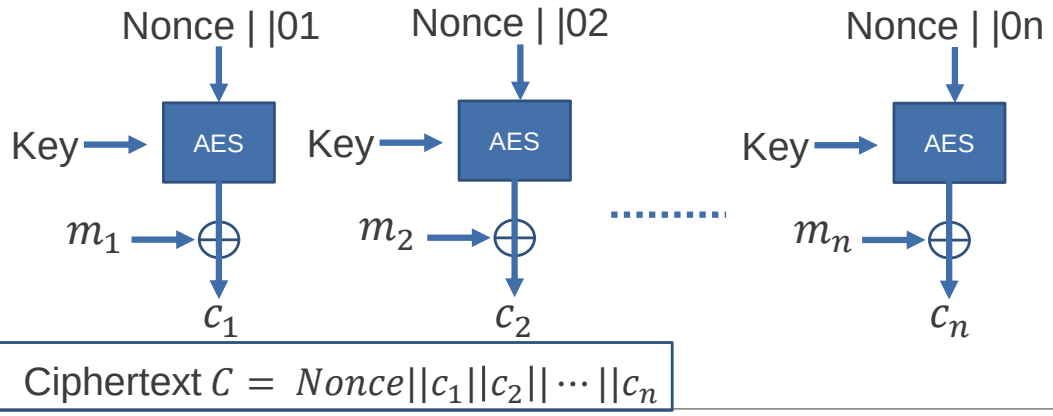
© 2023 Renesas Electronics Corporation. All rights reserved.

The reason why ECB mode gives this result is that ECB mode outputs the same ciphertext block when inputting the same plaintext block. Since the background is uniformly white in the original image, the background is encrypted to the same value, and the outline of the original image appears at the border between the background part and the others. To prevent such case, various block cipher modes of operation have been proposed, and these are considered to ensure that encrypted data blocks become different data even if the same plaintext blocks are continuously input.

EXAMPLE OF ENCRYPTION USE MODE: CTR MODE

CTR mode encryption

- Encrypts a nonce concatenated with counter by a block cipher (AES), and then computes exclusive OR (XOR) the results with the plaintexts.
- A nonce is concatenated to the ciphertext.
- Plaintext m is divided into blocks: $m = m_1||m_2||\cdots||m_n$.



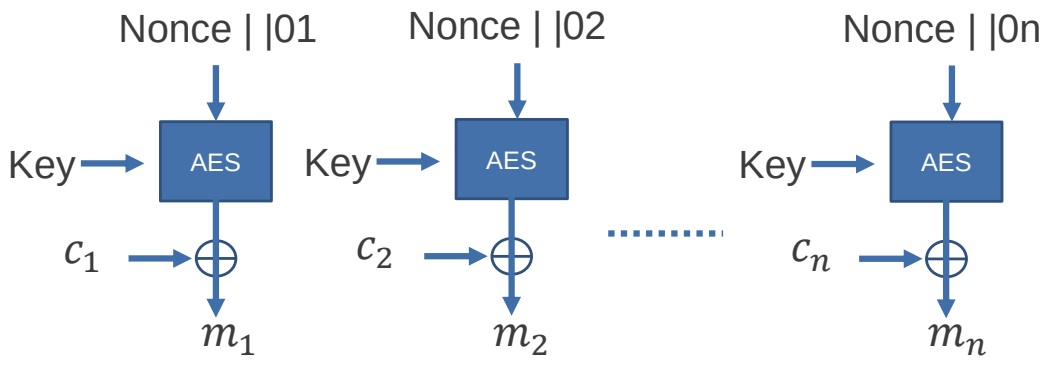
Ciphertext $C = Nonce||c_1||c_2||\cdots||c_n$

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains the solution of the previous page problem by using the block cipher mode of operation called CTR mode or counter mode. In CTR mode, plaintext m is separated by 128-bits blocks in the same way as well as ECB mode, but handling of inputs and outputs to AES are different. In CTR mode, instead of direct encryption of messages by AES, data that combines random numbers called nonce and a counter is encrypted with AES for each block. The result of this encryption and each plaintext block are XORed, and the data is used as the final ciphertext. Since the counters are incremented by one for each block, the output ciphertext differ for each block, and the ciphertext c1 and c2 encrypted in CTR mode differ even if the plaintext blocks m1 and m2 are the same value. By such method, the problem of the penguin image in the previous page can be avoided.

EXAMPLE OF ENCRYPTION USE MODE: CTR MODE

CTR mode decryption



Decryption functions of block cipher is not required.

© 2023 Renesas Electronics Corporation. All rights reserved.

This figure shows the decryption process in CTR mode. As with encryption, decryption process of CTR mode encrypts nonce and counters for each block with AES. By operating XOR to the encrypted messages, the plaintext can be obtained. In other words, only encryption is performed as AES primitive.

OTHER MODES OF OPERATION

- A number of block cipher modes of operations have been proposed.
- CRYPTREC Ciphers List recommends the following block cipher modes:
  - Encryption modes: CBC, CFB, CTR, and OFB.
  - Authenticated encryption modes: CCM, and GCM.

CRYPTREC website
https://www.cryptrec.go.jp/method.html

© 2023 Renesas Electronics Corporation. All rights reserved.

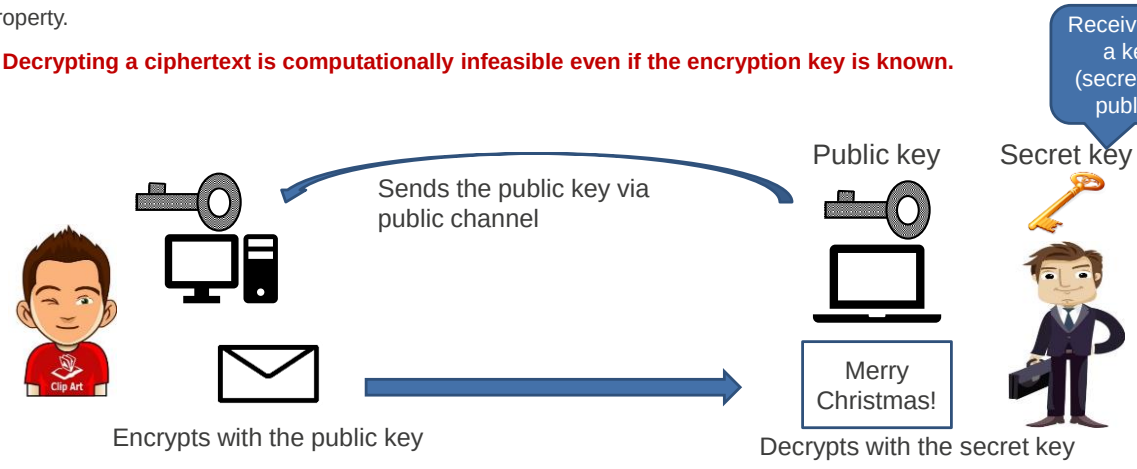
So far, the problems of ECB mode and countermeasures using CTR mode are explained, but various other block cipher modes of operation exist. Modes such as CBC, CFB, CTR, and OFB are used for message encryption. In addition, modes such as CCM and GCM are used for authenticated encryption. The authenticated encryption modes do not only encrypt messages, but also generate an authentication tag, which is used to check whether the messages are tampered.

PUBLIC KEY CRYPTOGRAPHY

© 2023 Renesas Electronics Corporation. All rights reserved.

PUBLIC KEY CRYPTOGRAPHY

- Public key cryptography is a cryptography which uses different keys for encryption and decryption and has the following property.
  - **Decrypting a ciphertext is computationally infeasible even if the encryption key is known.**



- A public key cryptography solves the key sharing problem because the public key can be sent via a public channel.
  - However, countermeasures against public key replacement attacks are required (PKI).

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains a public key cryptography, which uses different keys for encryption and decryption. The public key cryptography has the characteristic that decryption of ciphertext is computationally infeasible even if the encryption key is known. The reason why computational complexity is a condition is because it is theoretically possible to decrypt the ciphertext if the public key is known, but the decryption takes hundreds or thousands of years even if modern computers are used. This characteristic makes it possible to realize applications like this illustration. First, this illustration shows the flow in which the left person encrypts a message and sends it to the right person. The public key can be sent via the public channel because there is no problem in having third parties know the public key by the previous condition regarding computational complexity. Then, the sender encrypts the message with the public key and sends it to the receiver. Finally, the receiver decrypts the encrypted message with the secret key. However, there is one important point to note. There is a threat that the ciphertext could be decrypted eventually by replacement of the public key during the key exchange process.

ALGORITHMS OF PUBLIC KEY CRYPTOGRAPY

Public key cryptography consists of three algorithms

- Key generation algorithm
  - Inputs a security parameter 1^k and outputs a key pair (a secret key and a public key).
  - $Gen(1^k) \rightarrow (sk, pk)$
- Encryption algorithm
  - Inputs the public key and a plaintext and outputs a ciphertext.
  - $Enc(pk, m) \rightarrow c$
- Decryption algorithm
  - Inputs the secret key and the ciphertext and outputs the plaintext.
  - $Dec(sk, c) \rightarrow m$

© 2023 Renesas Electronics Corporation. All rights reserved.

The public key cryptography consists of three algorithms. A key generation algorithm is the algorithm to generate a key pair including a secret key and a public key. An encryption algorithm is the algorithm to encrypt a plaintext with a public key. A decryption algorithm is the algorithm to decrypt a ciphertext to the plaintext with a secret key.

RSA ENCRYPTION

- RSA encryption is the first public key cryptography proposed by R. Rivest, A. Shamir, and L. Adleman in 1977.
- Security of RSA is based on difficulty of factoring large numbers.
- It is still widely used nearly 50 years after its invention.

R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun, ACM 21(2): 120-126, 1978.

© 2023 Renesas Electronics Corporation. All rights reserved.

This page introduces RSA encryption which is a famous public key cryptography. The RSA encryption is the first public key cryptography proposed by R. Rivest, A. Shamir, and L. Adleman in 1977. Security of RSA is based on difficulty of factorization of large numbers. The RSA encryption is still widely used nearly 50 years after its invention.

MODULAR EXPONENTIATION

The RSA algorithms use modular multiplications.

The following is a brief review of modular multiplications.

- A remainder of a division of a by p is denoted by $a \ mod \ p$.
- For the integer a, an integer b which satisfy $a \times b \ mod \ p = 1$ is called a modular multiplicative inverse of a and denoted by $b = a^{-1} \ mod \ p$.

✓ For an integer a, the necessary and sufficient condition for the existence of the inverse is to hold $\gcd(a, p) = 1$.

✓ An inverse $a^{-1} \ mod \ p$ can be efficiently computed by using the fact that $a^{p-2} = a^{-1} mod \ p$ holds. (Fermat's little theorem)

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains modular multiplications, which is often used in RSA algorithms, and an inverse element defined in modular multiplications. We will omit the detailed explanation.

ALGORITHMS OF RSA (1/2)

- Key generation algorithm
- Input: security parameter 1^k
- Output: key pair of a public key and a secret key

Procedure:

1. Create two large primes p, q and calculate $N = p \times q$.
2. Calculate $L = lcm(p-1, q-1)$.
3. Create two integers e and d, satisfying $ed = 1 \; mod \; L$.
   - To compute e and d, first generates e which is coprime to L, and then calculate $d = e^{-1} mod \; L$.
   - $e = 655537$ is often used.
4. Outputs the public key (e, N) and the secret key d.

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains procedures of the key generation algorithm of RSA. We will omit the detailed explanation.

RSA ALGORITHM (2/2)

- Encryption algorithm
- Input: a plaintext $m(1 \leq m < N)$, a public key (e, N)
- Output: a ciphertext c

Procedure:

1. Calculate $c = m^e \ mod \ N$.
2. Output the ciphertext c.

- Decryption algorithm
- Input: a ciphertext c ,a secret key $d,$ and a public key N.
- Output: a plaintext m

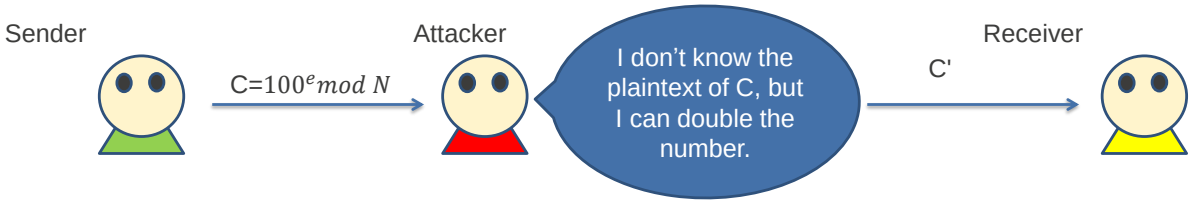
Procedure:

1. Calculate $m = c^d \ mod \ N$.
2. Output the plaintext m.

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains procedures of the encryption and decryption algorithms of RSA. We will omit the detailed explanation.

PLAIN RSA IS NOT SECURE



The receiver's public key (e, N) encrypts a plaintext 100.
$100^e mod\ N$

The attacker also knows the receiver's public key (e, N).
✓ Calculates $2^e mod\ N$.
✓ Calculates $C \times 2^e mod N$.
$C' = C \times 2^e mod\ N$
$= 100^e \times 2^e mod N = 200^e mod\ N$

Obtains 200 by decrypting C'.
The ciphertext was modified on the communication channel!

- The attacker can modify the ciphertext to any multiple, although the attacker does not know the value of the send data.
- In addition to this problem, plain RSA has another bad feature that the same ciphertexts are output if the same plaintexts are encrypted.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

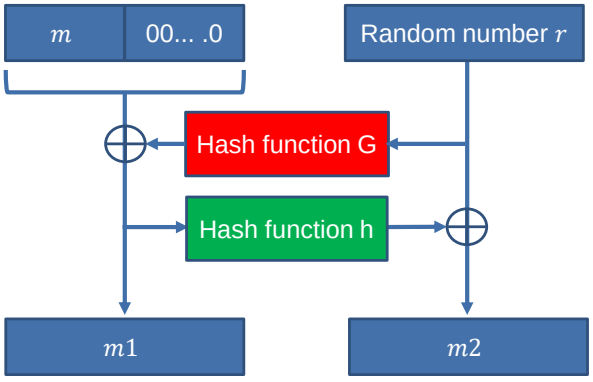
The RSA primitive is known to insufficient for security if it as is implemented. The implementation includes various problems. For example, this page shows how an attacker attacks the transmission of a number from the sender to the receiver. In RSA encryption, the attacker can modify the plaintext to any multiples without decryption of the ciphertext. In this example, a damage has occurred where the original value of 100 is modified to 200. Additionally, when the same key is used, the same ciphertexts are output if the same plaintexts are encrypted, which is a problem because the attacker can identify that the same plaintexts has been input.

FOR SECURE USE OF RSA (PADDING)

OAEP : Optimal Asymmetric Encryption Padding

M. Bellare et al., "Optimal asymmetric encryption – How to encrypt with RSA," LNCS 950, 1995.

- Security of RSA encryption is improved by transforming plaintexts.
- OAEP is standardized PKCS#1 ver. 2.0 and RFC 8017.



Encryption

1. Obtains $m1$ and $m2$ by transforming the plaintext m as shown in the left figure.
2. Concatenates $m1||m2$ and encrypts it with RSA.

Decryption

1. Obtains $m1||m2$ by decrypting the ciphertext.
2. Calculates $r' = H(m1) \oplus m2$.
3. Calculates $m' = G(r') \oplus m1$.
4. If the least significant bits of m' are not all zero, $\perp$ is output and decryption fails.
5. Outputs m.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

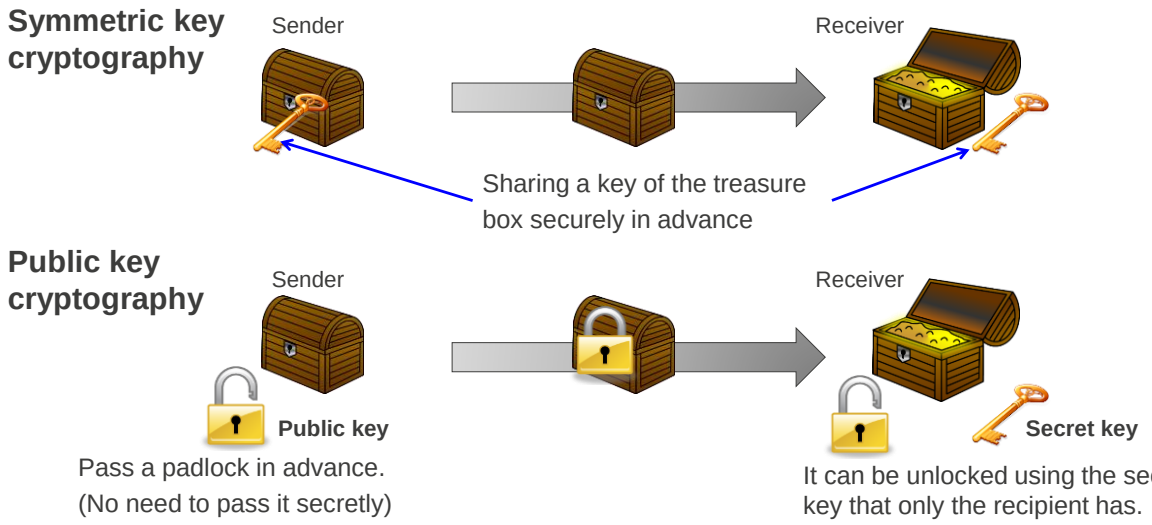
This page explains the method called OAEP padding, which is proposed for securely using RSA. OAEP is the method that is standardized PKCS#1 version 2.0. The OAEP prepares an input message m padding it with 0, and random numbers r of the same length as the padded input message. The input message is randomized by a hash function and XOR. As explained on the previous page, the problem that encryption of the same messages output the same ciphertexts is solved by the randomization of the input message. We will omit the detailed explanation, but the modification of the ciphertext can be detected by the fourth step of the decryption procedure described to the right.

HYBRID ENCRYPTION

© 2023 Renesas Electronics Corporation. All rights reserved.

SYMMETRIC KEY AND PUBLIC KEY CRYPTOGRAPHY

Symmetric key and public key algorithms can be considered as the analogy of the treasure box.



© 2023 Renesas Electronics Corporation. All rights reserved.

So far, we have explained the symmetric key and the public key cryptography. These concepts can be considered as an analogy of locking a treasure box like this illustration. The symmetric key cryptography can be considered as an analogy of treasure box with a keyhole, where the same key is used to open and lock the treasure box. The public key cryptography can be considered as an analogy of a padlock. In this case, anyone can lock the padlock, but only a person who has the key can open it.

COMPARING SYMMETRIC KEY AND PUBLIC KEY CRYPTOGRAPHY

Symmetric key cryptography

Features:
- An encryption key and a decryption key are the same.
- Encryption and decryption are generally fast.
- Key sharing problem.

Famous algorithms:
- Block ciphers
  - AES, DES, MISTY, KASUMI, Camellia
- Stream ciphers
  - ChaCha20, Multi-S01, RC4, KCipher-2

Public key cryptography

Features:
- An encryption key and a decryption key differ. The encryption key does not contain secret.
- Encryption and decryption are generally slow.
- Relatively easy to share the key.

Famous algorithms:
- RSA, ElGamal encryption, Cramer–Shoup encryption.

© 2023 Renesas Electronics Corporation. All rights reserved.

This page summarizes advantages and disadvantages of symmetric key cryptography and public key cryptography. The blue texts represent the advantages, and the red texts represent the disadvantages. The symmetric key cryptography has features, which are fast but difficult to share a key. The public key cryptography has features, which are easy to share a key but slow in processing.

HYBRID ENCRYPTION

Symmetric key cryptography and public key cryptography have both advantages and disadvantages

Symmetric key cryptography: Processing is fast. Key sharing problem exists.
Public key cryptography: Processing is slow. Easy to share the key.

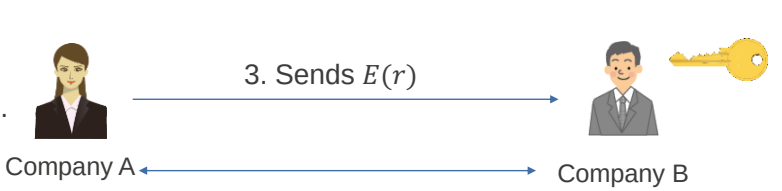
Combining symmetric key cryptography and public key cryptography can make another public key cryptography that can achieve fast processing.

© 2023 Renesas Electronics Corporation. All rights reserved.

As explained on the previous page, the symmetric key cryptography, and the public key cryptography each have the advantages and the disadvantages. To solve the disadvantages of both cryptographies, a method called hybrid encryption have been proposed, which combines a symmetric key cryptography and a public key cryptography.

HYBRID ENCRYPTION

Sharing the key of symmetric cryptography by using a public key cryptography, and encrypting messages by symmetric key cryptography.



Blue: Processing in public key cryptography
Red: Processing in symmetric key cryptography

High-speed encryption and decryption are possible because encryption and decryption of messages are processed by symmetric key cryptography.
The fast public key cryptography can be constructed by combining symmetric key crypto and public key crypto.

© 2023 Renesas Electronics Corporation. All rights reserved.

The procedures are explained by this illustration. First, the company A generates a random number r, and encrypts r with the company B's public key, and obtains E(r). E(r) is sent to the company B. Next, the company B decrypts E(r) with the company B's secret key and obtains r. From the above, the companies A and B can secretly share the random number r. Subsequent communications are encrypted with the random number r and a symmetric key cryptography such as AES. This method becomes fast as a whole processing because subsequent communication data can be encrypted and decrypted with a high-speed algorithm although the initial process of decryption and encryption of the random number r takes time.

KEY EXCHANGE PROTOCOL

© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

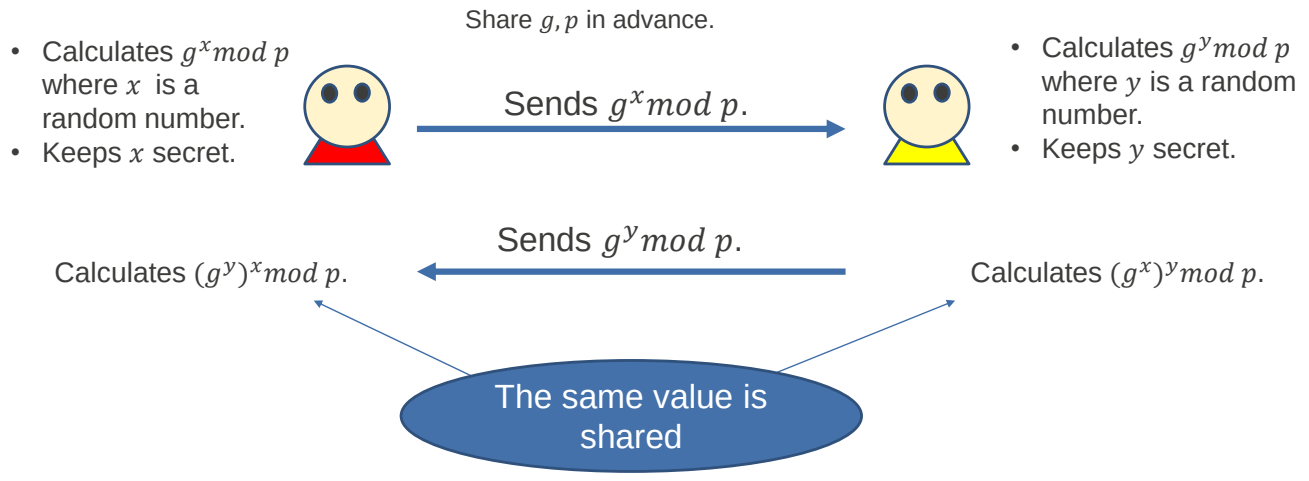
DIFFIE-HELLMAN KEY EXCHANGE PROTOCOLS

- This algorithm becomes the basis for the idea of public key cryptography.
- This method shares secret information (key) without secret communication channels.
- This is proposed in 1976.
  - W. Diffie, and M. E. Hellman, “New Directions in Cryptography,” IEEE Trans. on Information Theory, 1976.
- This method uses difficulty of discrete logarithm problem as the rationale of security.
  - Discrete logarithm problem: This is a problem to find x when g and y, that satisfy $g^x = y$, are given in $mod\ p$, such as explained on the RSA pages.

© 2023 Renesas Electronics Corporation. All rights reserved.

Several key exchange protocols exist in addition to the method explained in the section of the hybrid encryption. Its representative method is a Diffie-Hellman key exchange protocol.

DIFFIE-HELLMAN KEY EXCHANGE PROTOCOLS



Since the shared value is biased (not a perfect random number), an unbiased value generated by a Key Derivation Function (KDF) can be used as cryptographic key.

© 2023 Renesas Electronics Corporation. All rights reserved.

In procedures of the Diffie-Hellman key exchange protocol, two persons who would like to share a key, respectively, generate random numbers x and y secretly, calculate g^x mod p and g^y mod p, and exchange the results with each other. Finally, the persons calculate g^(xy) mod p respectively by x and g^y mod p or y and g^x mod p. By this method, the persons can share the value of g^(xy) mod p without the disclosure of x and y.

SECURITY OF DIFFIE-HELLMAN KEY EXCHANGE PROTOCOL

- Calculates $g^x mod\ p$ where x is a random number.
- Keeps x secret.

Sends $g^x mod\ p.$

- Calculates $g^y mod\ p$ where y is a random number.
- Keeps y secret.

Sends $g^y mod\ p.$

Calculates $(g^y)^x mod\ p.$

Calculates $(g^x)^y mod\ p.$

- Cannot calculate x from $g^x mod\ p.$
- Cannot calculate y from $g^y mod\ p.$

⇒ Cannot calculate g^{xy}.

Difficulty of the discrete logarithm problem

Attacker, who is looking at the channel

However, it is not known if it is possible to break the DH key exchange without solving the discrete logarithm problem.

© 2023 Renesas Electronics Corporation. All rights reserved.

The Diffie-Hellman key exchange protocol needs to send g^x mod p or g^y mod p to a communication partner, as supposing an attacker eavesdrops on the numbers. However, deriving g^(xy) mod p from g^x mod p or g^y mod p is considered difficult in terms of computational complexity. A problem that calculates x from g^x mod p is called the discrete logarithm problem whose difficulty is the basis of security of the Diffie-Hellman key exchange protocol.

KEY EXCHANGE PROTOCOL AND FORWARD SECRECY

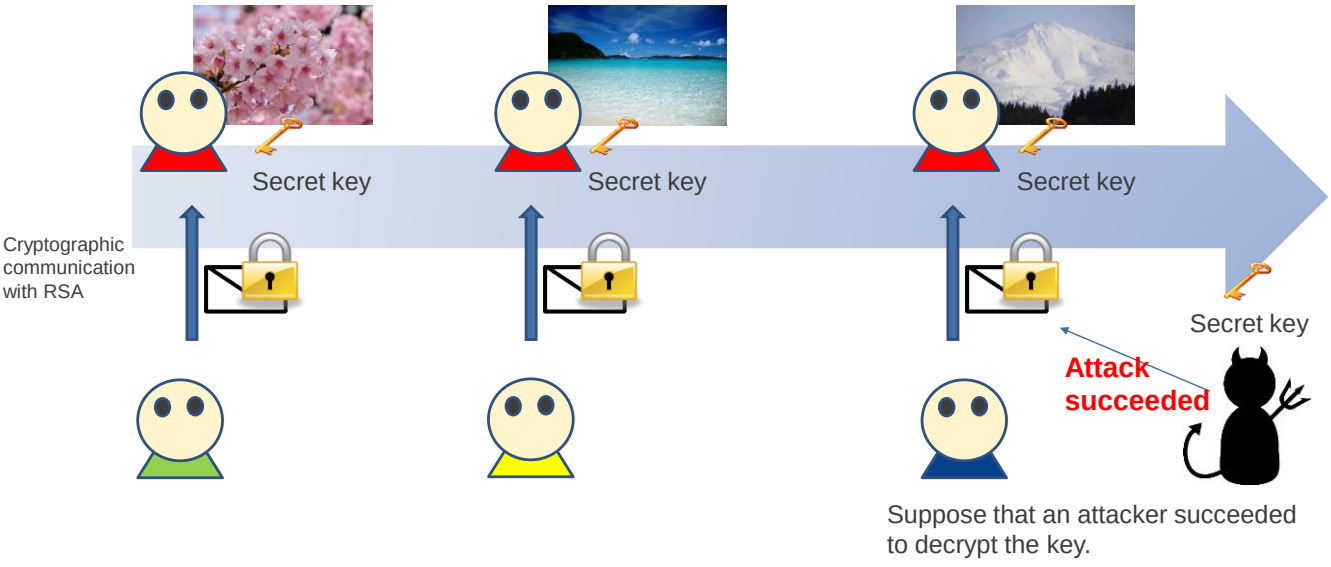
- We introduced two methods to share keys:
  - Public key cryptography (e.g. RSA).
  - Diffie-Hellman key exchange protocol.
- **Key exchange mechanisms with static RSA or Diffie-Hellman have been removed from TLS 1.3.**

  The reason is that they do not provide Forward Secrecy.

© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

Until here, we have explained the key sharing method using RSA in the section of the hybrid encryption, and the Diffie-Hellman key exchange protocol. However, key exchange mechanisms with static RSA or Diffie-Hellman have been removed from TLS 1.3. The reason is that they do not provide Forward Secrecy.
In next slides, we will explain what Forward Secrecy is.

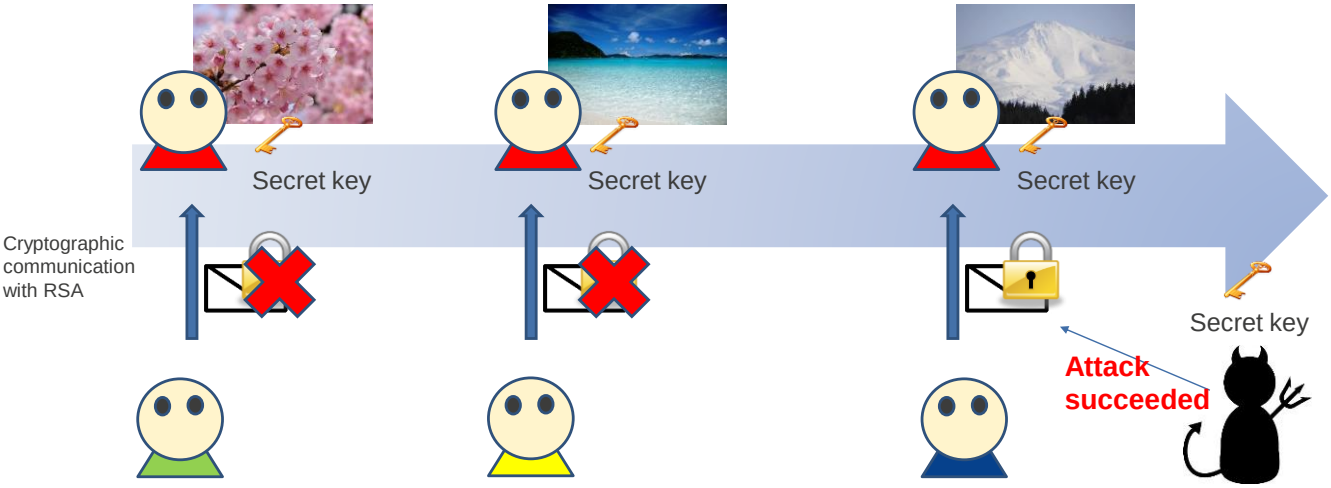
FORWARD SECRECY



© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

This page shows an example of cryptographic communications using a shared key during a certain period. This figure supposes that an attacker breaks an RSA encryption at a point and succeeds in obtaining a private key illegally.

FORWARD SECRECY

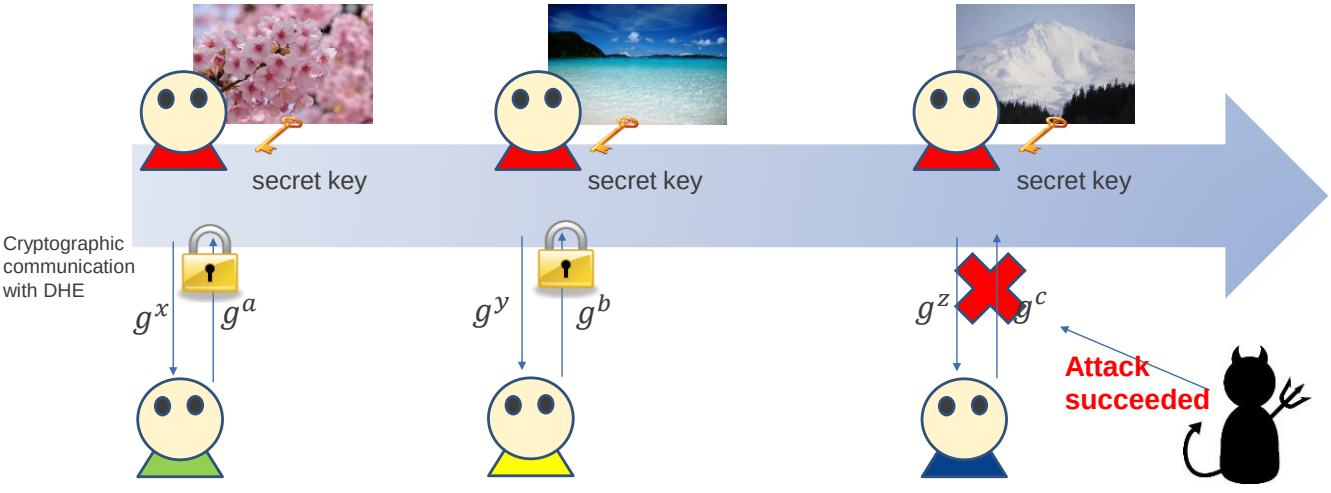


If the attacker succeeds to decrypt the ciphertext at some point, all previous communication contents will be decrypted (suppose that the attacker was collecting the ciphertexts).

© 2023 Renesas Electronics Corporation. All rights reserved.

In this case, the attacker can decrypt all the encrypted messages by the known secret key and identify all the communication contents. The next page explains Forward Secrecy, which has a feature that previous communication contents cannot be decrypted even if a key at a point is leaked.

FORWARD SECRECY



Even if the protocol is broken at some point, the confidentiality of communications before that point is kept secure (forward secrecy).

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

A method to secure Forward Secrecy with a key exchange protocol is to randomly change a secret key used by the Diffie-Hellman key exchange protocol every time. By this method, other messages cannot be decrypted because each communication uses different secret keys to share keys even if the attacker obtains the secret key used for the key exchange protocol at a point. Such temporary random key is called the Ephemeral Key, and the Diffie-Hellman key exchange protocol with the Ephemeral Key is abbreviated as DHE.

CRYPTOGRAPHIC HASH FUNCTION

© 2023 Renesas Electronics Corporation. All rights reserved.

The section explains cryptographic hash function, which is a cryptographic primitive used in cryptographic applications. The hash function outputs computing results of fixed length when data of any length are input. This computing result is called a hash value or a digest. In addition to this feature, a hash function also has features explained on the next page. The hash function is most often used as a component of other cryptographic algorithms. For example, Digital signature, Message Authentication Code, and Key Derivation Functions use the hash function.

CRYPTOGRAPHIC HASH FUNCTION

- A hash function maps data of arbitrary length to fixed length data (called a hash value or digest).
- A hash function designed for cryptographic purposes is called a **cryptographic hash function.**

Arbitrary length bit string

Fixed-length bit string (hash value)

Hash functions are used as a component of cryptographic algorithms (Public key cryptography, digital signature, MAC(Message Authentication Code), Key Derivation Functions (KDF), etc.).

© 2023 Renesas Electronics Corporation. All rights reserved.

The section explains cryptographic hash function, which is a cryptographic primitive used in cryptographic applications. The hash function outputs computing results of fixed length when data of any length are input. This computing result is called a hash value or a digest. In addition to this feature, a hash function also has features explained on the next page. The hash function is most often used as a component of other cryptographic algorithms. For example, Digital signature, Message Authentication Code, and Key Derivation Functions use the hash function.

SECURITY OF CRYPTOGRAPHIC HASH FUNCTIONS

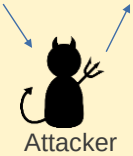
- The following security notions are defined for cryptographic hash functions.
- Each security notion is defined as follows:

Pre-image resistance

It is hard to find a pre-image of the randomly given hash value.

A random hash value y is given

Computes a value x that satisfies



Attacker

Second pre-image resistance

Given a random message, and it is hard to find another message that hash to the same value.

A random message z is given

Computes a message w which satisfies $w \neq z$ and $Hash(w) = Hash(z)$.

Attacker

Collision resistance

It is hard to find two messages that hash to the same value.

Computes two different messages x, y which satisfy Hash(x)=Hash(y).

Attacker

Difficulty of successful attack

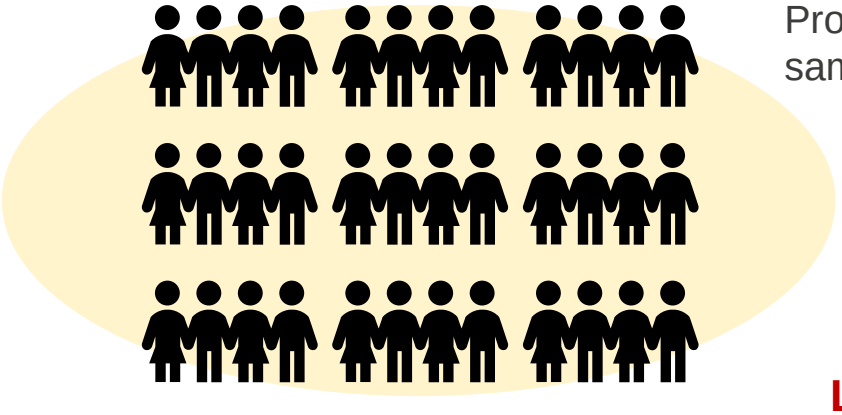
$O(2^n)$ $O(2^n)$ $O(2^{n/2})$

n: Output length

© 2023 Renesas Electronics Corporation. All rights reserved.

The security notion of the hash function is defined with the following three items. A pre-image resistance is the feature that it is hard to find a message of any given hash value. A second pre-image resistance is the feature that it is hard to find another message that hashes to the same value when a random message given. A Collision resistance is the feature that it is hard to find two messages that hash to the same value. When different messages return the same hash value, it is called a collision. Among the three security notions for the hash function, collision resistance is the most easily broken feature.

REASON WHY COLLISION RESISTANCE IS EASIER TO BREAK: BIRTHDAY PARADOX



36 members

Probability of having the same birthday pair

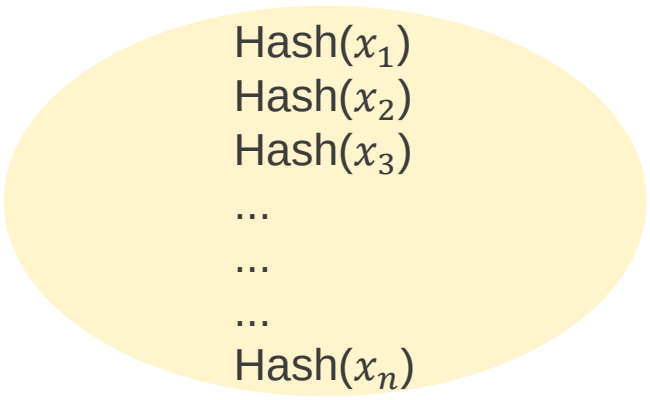
0.832

Looks higher than expected

© 2023 Renesas Electronics Corporation. All rights reserved.

The Birthday paradox exists for an example of ease of breaking the collision resistance. It expresses that a probability of having the same birthday pair among 36 members is 0.832 and that the probability looks higher than expected.

REASON WHY COLLISION RESISTANCE IS EASIER TO BREAK: BIRTHDAY PARADOX



The probability that a pair of the same hash values exists in this group (in other words, collision) is also higher than expected.

In the case of an n-bit output hash, if $1.1774 \times 2^{\frac{n}{2}}$ hash values are computed, the probability that a pair of the same hash values exists in that is 0.5.

© 2023 Renesas Electronics Corporation. All rights reserved.

Supposing that a hash function outputs n bits hash value, it is known that a probability of existence of a pair of the same hash values is 0.5 if hash values of 1.1774 * 2^(n/2) messages are collected.

HASH LENGTH AND SECURITY

In the case of an n-bit output hash, if an attacker computes

$$1.1774 \times 2^{\frac{n}{2}}$$

hash values, the attacker can find a collision pair with the probability 0.5.

For example, SHA-1's output length is 160 bits. A collision can be found with a probability of 0.5 if 2^{80} hash values are computed.

We say that collision resistance against SHA-1 has 80bit security.

A hash function with an n bits output length has $n/2$ bit security.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

For example, if 2^80 hash values generated from SHA-1 with 160-bits output length are collected, the probability of having a pair of the same hash value is 0.5. In other words, an attacker can break the collision resistance with the probability of 0.5 by computing 2^80 hash values. The security strength of this case is expressed as having an 80-bits security.

MAJOR HASH FUNCTIONS

Algorithm	Hash length (bit)	Internal structures	Security
MD5	128	MD(Merkle-Damgaard)	Collision resistance was broken
SHA-1	160	MD(Merkle-Damgaard)	Collision resistance was broken
SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256)	224, 256, 384, 512 These depend on the methods.	MD(Merkle-Damgaard)	Secure
SHA-3 (SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256)	224, 256, 384, 512, variable (SHAKE)	Sponge	Secure

© 2023 Renesas Electronics Corporation. All rights reserved.

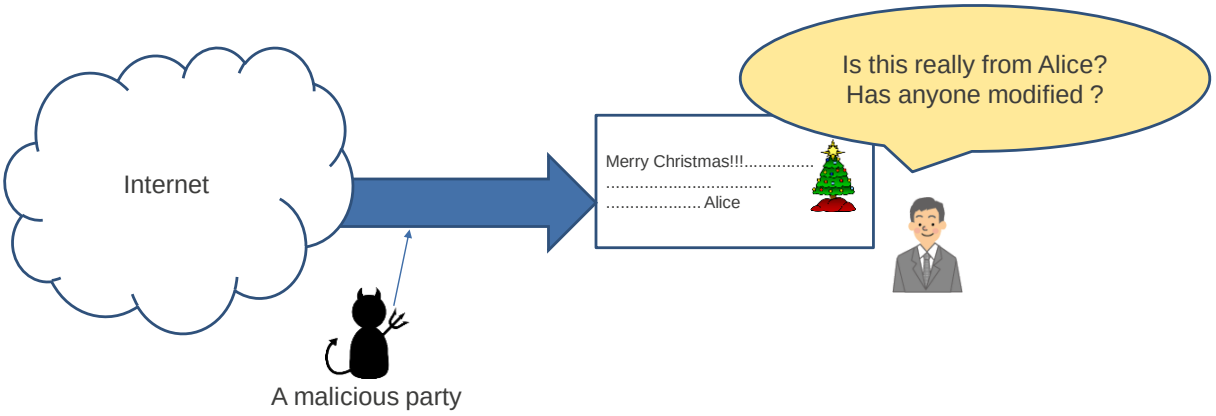
This page lists representative hash functions. MD5 and SHA-1 are not recommended because their collision resistance is broken with current technology. Currently, the SHA-2 family and the SHA-3 family are considered safe, and the adoption of the SHA-3 family is increasing because it has the same security strength as the SHA-2 family but can compute faster.

MESSAGE AUTHENTICATION CODE
DIGITAL SIGNATURE

© 2023 Renesas Electronics Corporation. All rights reserved.

DIGITAL SIGNATURE AND MESSAGE AUTHENTICATION CODE (MAC)

Digital Signature and Message Authentication Code (MAC) are technologies to verify that the data is **created by an intended party** and is not **modified by a malicious party.**
Unlike cryptography, these are not technologies against eavesdropping.

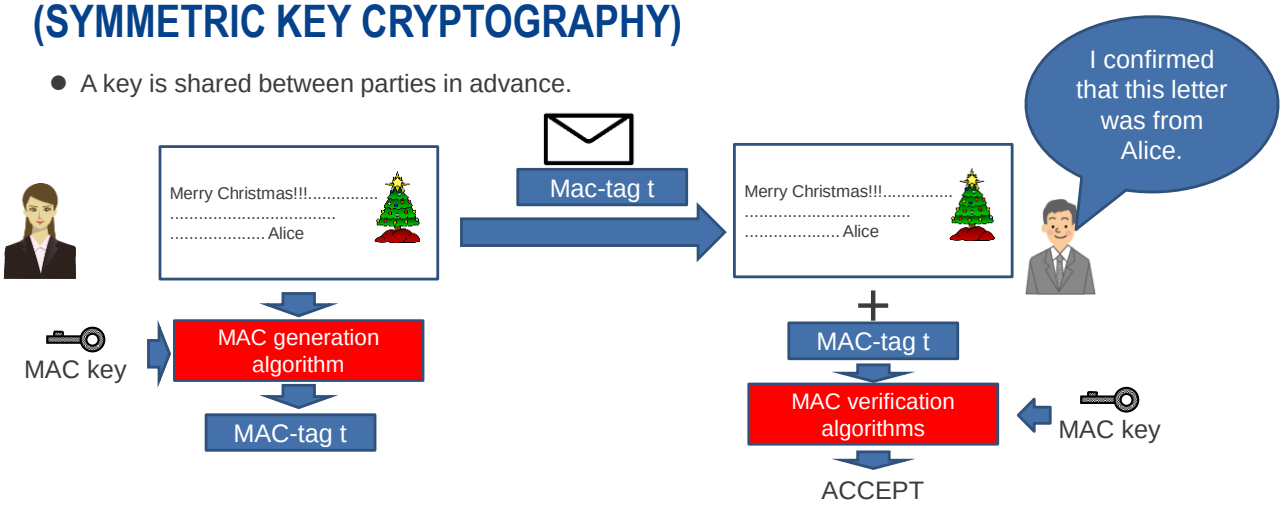


© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

This section explains Message Authentication Code called MAC, and the digital signature, which are technologies to verify the integrity of a message. These technologies confirm whether data were really generated by an intended party and the data have been tampered with by anyone. As a reminder, it is necessary to encrypt the data that you do not want to be eavesdropped by encryption algorithms since these technologies do not perform the encryptions.

MESSAGE AUTHENTICATION CODE (SYMMETRIC KEY CRYPTOGRAPHY)

- A key is shared between parties in advance.



- Major MAC Algorithms: HMAC, CMAC
- In the case that encryption and MAC are used, first encrypt the plaintext and then generates MAC on the ciphertext.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

First, the followings explain MAC. MAC is a one of symmetric key cryptography and can verify a message among parties sharing a MAC key. In this illustration, a left person inputs a message and a MAC key into a MAC generation algorithm and generates a MAC tag and sends the MAC tag and the message to a right person. The MAC tag is used to prove the integrity of the message. The right person inputs the message, the MAC tag, and the MAC key into the MAC verification algorithm and verifies the integrity of the message. If a message sender is not an intended person, the person cannot generate an intended MAC tag because the person does not have the MAC key. Additionally, when the message sent from the intended person was modified during communications, the MAC algorithm can detect the modification because the received MAC tag differs from a MAC tag generated from a received message. As representative MAC algorithms, there are HMAC using a hash function, and CMAC using a block cipher such as AES.

SEQUENCE OF MAC AND ENCRYPTION

- ◎Encrypt then MAC: The plaintext is encrypted, then a MAC is generated on the ciphertext. (Recommend)

  - It is proven to have strong security.

- ○MAC then encrypt: A MAC is generated on the plaintext and the plaintext and MAC are encrypted.

  - It is not proven to have strong security in general.

- × Encrypt and MAC: A MAC is generated on the plaintext, and the plaintext is encrypted without the MAC.

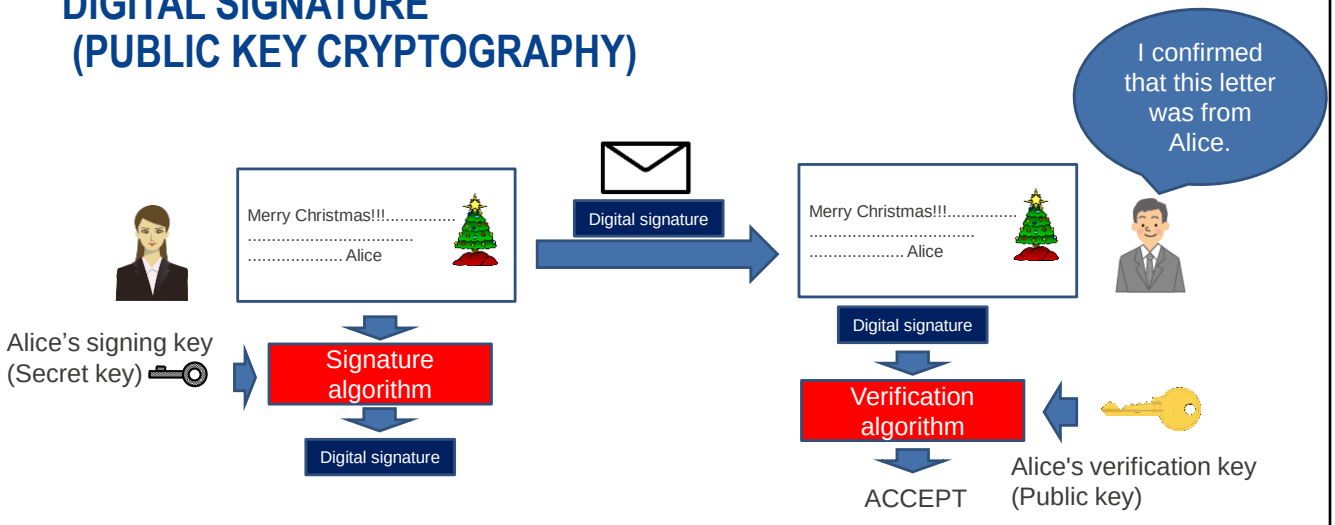
  - Some information about the plaintext may be disclosed from the MAC.

C1 = Enc(k1, "pay $100") | | MAC (k2, “$100 paid")
C2 = Enc(k1, "pay $100") | | MAC (k2, “$100 paid")
MAC parts of C1 and C2 indicates that the plaintexts of C1 and C2 are the same..

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

Since the MAC cannot prevent eavesdropping、generating a MAC and encryption may be performed against a message. At this time, it is necessary to consider whether to generate a MAC with a ciphertext or to encrypt both a plaintext and a MAC generated by the plaintext. As a recommendation, it is better to generate the MAC against the ciphertext from a security perspective. An upper figure corresponds to this case. A middle figure corresponds to a case that encrypts both a plaintext and the MAC computed from the plaintext. This case is not recommended from the security perspective because it requires inputting the unverified ciphertext into a decryption function. A lower figure is a case where encrypts a message and generate a MAC from a plaintext. However, this case has a problem that can cause leakage of the plaintext because if the MACs have the same value, it can be estimated that their plaintexts are the same value. This problem affects security when the same messages appear frequently and are sensitive.

DIGITAL SIGNATURE (PUBLIC KEY CRYPTOGRAPHY)



- The signature is created by the signing key (secret key) and verified by the verification key (public key).
- Since the verification key can be public, anyone can verify the signature.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

The followings explain a digital signature. It can be roughly considered as an algorithm that a processing like MAC is performed by public key cryptography. In the digital signature, the integrity of a message is verified with its signature. The signature can be generated with a signing key and verified with a verification key. Since the verification key is public, anyone can verify the signature. To operate the digital signature, a person sending the message is required to generate a key pair in advance and prove that its verification key is generated by the person. The public key algorithms used for the digital signature include methods such as an RSA signature and ECDSA.

EXAMPLES OF DIGITAL SIGNATURE: RSA SIGNATURE

RSA signature is one of the most widely used digital signature schemes.

- Key generation algorithm:

Outputs a verification key (e, N) and a signing key d.

- Signing algorithm: takes a message m, and the signing key (d, N) as input.

Computes $h(m)$ by using the cryptographic hash function h and computes $\sigma = h\ (m)^d mod\ N$.

Outputs σ.

- Verification algorithm: takes the signature σ, message m, and the verification key (e, N) as input.

Computes $h(m)$ by using the cryptographic hash function h and compares $\sigma^e mod\ N$.

If $h(m) = \sigma^e mod\ N$ holds, outputs ACCEPT, otherwise REJECT.

© 2023 Renesas Electronics Corporation. All rights reserved.

This page explains an algorithm of an RSA signature. The digital signature algorithm includes a key generation algorithm, a signature algorithm, and a verification algorithm. These algorithms are shown here.

RSA SIGNATURE

- RSA signature is the first digital signature algorithm.
- Since key length and signature length of 2048 bits or more are required to ensure security, it might be difficult to use on embedded devices due to restricted memory space.

© 2023 Renesas Electronics Corporation. All rights reserved.

The RSA signature algorithm is the world's first proposed method in digital signatures. Since key length and signature length of 2048 bits or more are required to ensure security, it may be difficult for use of embedded devices due to restricted memory space. However, recent products are increasingly using 3072 or 4096-bits key lengths.

RSA SIGNATURE PADDING

- For RSA signatures, a padding is used to enhance security as well as RSA cryptography.
  - The followings are the major padding methods.
    - RSA-PKCS#1
    - RSA-PSS

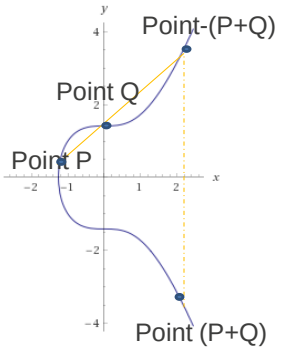
© 2023 Renesas Electronics Corporation. All rights reserved.

In the explanation of the RSA encryption, we explained that the security problem is caused if the RSA primitive as is implemented. The same problem also exists in the RSA signature. These padding methods were proposed to securely implement the RSA signature. Therefore, you should use these padding methods when you will use the RSA signature.

DIGITAL SIGNATURE USING ELLIPTIC CURVE

- Key length of RSA signature
  - Long key length: 2048 bits or longer public key N are required.
  - Currently, the world record for prime factorization of a natural number is 768 bits.

- Elliptic Curve Cryptography
  - A cryptographic algorithm designed on an elliptic curve.
    - ✓ Elliptic curve: $y^2 = x^3 + ax + b$.
    - ✓ On the elliptic curve, an addition with two rational points can be defined.
  - **Since efficient attack method for the discrete logarithm problem on elliptic curves is not found**, security is ensured even with short key lengths.
  - It is estimated that elliptic curve cryptography can achieve the same level of security as 2048-bits RSA with a 224-bits key.



Graph of $y^2 = x^3 + 2$ and additions of points

© 2023 Renesas Electronics Corporation. All rights reserved.

A digital signature algorithm with an elliptic curve also exists. This page explains the key length of the RSA signature and the Elliptic Curve Cryptography. The most important aspect of the elliptic curve digital signature is that it can ensure the same security strength with a shorter key length than the RSA signature. The elliptic curve cryptography can achieve the same level of security as 2048-bits RSA with a 224-bits key.

ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHMS

- EC-Schnorr Signature (Elliptic curve version of Schnorr signature)
- ECDSA (Digital signature that realizes DSA on elliptic curves)
- EdDSA (Ed25519, Ed448)

© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

EC-Schnorr, ECDSA, EdDSA, and so on exist as methods of the digital signature with the elliptic curve.

USE OF DIGITAL SIGNATURE AND MAC

Difference between digital signature and MAC

Digital signature has three features that the MAC does not have.

1. Public Verifiability: Anyone can verify a signature because the verification key can be public.
2. Transferability: A recipient of a message and a signature can prove the validity of the signature to third parties.
3. Non-repudiation: Once a signer generates and publishes a signature, the generated signature cannot be repudiated.

It is better to use MAC if the following conditions are met.

1. The above three features of a digital signature do not be required (or do not want to have).
2. Key sharing is feasible.

© 2023 Renesas Electronics Corporation. All rights reserved.

The MAC and the digital signature were explained until here. This page explains a summary of the difference between the digital signature and the MAC. The digital signature has three features that the MAC does not have. Public Verifiability means that anyone can verify a signature because the verification key can be public. Transferability means that a recipient of a message and a signature can prove the validity of the signature to third parties. It depends on a characteristic that the verification key can be public. Non-repudiation means that once a signer generates and publishes a signature, the generated signature cannot be repudiated because only the signer who has the signing key can create a valid signature of a message. In the case of the MAC, a verifier also has a verification key, so the verifier can also generate the MAC value. If these three features are not required and a sender and a recipient of a message can share a secret key, it is recommended to use the MAC.

RANDOM NUMBER GENERATION

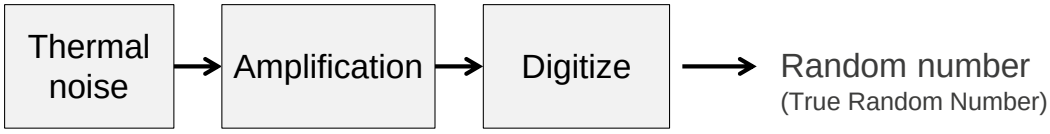
© 2023 Renesas Electronics Corporation. All rights reserved.

RANDOM NUMBER GENERATION

Random numbers are used for cryptographic key generations or in cryptographic algorithms.
Random numbers of low-quality compromise security of the cryptographic schemes.

■ True random number generator
⇒ A device that generates random numbers using physical phenomena.
A dedicated circuit is required. It is difficult to speed up.

Example



*There is randomness test programs which evaluate the quality of random number generators.

© 2023 Renesas Electronics Corporation. All rights reserved.

The explanation of cryptographic algorithms ends on the previous page. The following explains random number generation, which is important for cryptographic algorithms. A random number is used to generate a random key and is used inside cryptographic algorithms. A random number used in these cases must be high-quality because security of cryptography may be compromised. The high-quality random number are generated by a true random number generator, which is the device that outputs a random number by converting conversion from physical information such as thermal noise within a chip into a digital value. This processing requires a dedicated circuit and is difficult to fast.

RANDOM NUMBER GENERATION

■ Pseudo-random number generator (PRNG)

- PRNG is an algorithm which generates a longer random number sequence from a short random number called a seed. High-speed processing is possible.

Note: For a cryptographic use, a pseudo-random number generator designed for cryptographic purpose shall be used.
For example, the pseudo-random number generator defined by NIST SP 800-90A, etc.

- A true random number can be used as a seed of a PRNG.
- Once a seed is known to third parties, the third party can compute the random number sequence generated from the seed. Seeds must be kept secret.
- A random number sequence generated by a cryptographic PRNG is computationally indistinguishable from a true random number sequence. Therefore, random numbers generated by a cryptographic PRNG can be securely used for cryptographic purpose.

© 2023 Renesas Electronics Corporation. All rights reserved.

A pseudo random number generator is used with the true random number generator. A pseudo random number generator inputs an initial value called 'Seed' into the algorithm and outputs a calculation result as a random number. The pseudo random number generator is faster than the true random number generator. Therefore, it is used to obtain a long random number by the pseudo random number generator with a seed, which is a short random number and initially generated by the true random number generator. A pseudo random number generator is also called a deterministic random bit generator because it always outputs the same random number if the seed is the same. This feature also means that if the seed value is leaked to an attacker, the output random number can be guessed. Therefore, the seed of the pseudo random number generator must be managed not to be leaked to a third party. The random numbers generated by a pseudo random number generator can be securely used for cryptographic purposes as long as the seed is sufficiently random, and the algorithm defined in NIST SP800-90A is used.

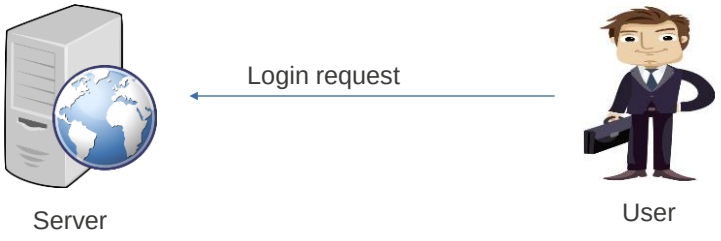
AUTHENTICATION PROTOCOL

© 2023 Renesas Electronics Corporation. All rights reserved.

AUTHENTICATION PROTOCOL

The purpose of an authentication protocol is to identify the communication peer.

Example: authentication for user login to the server

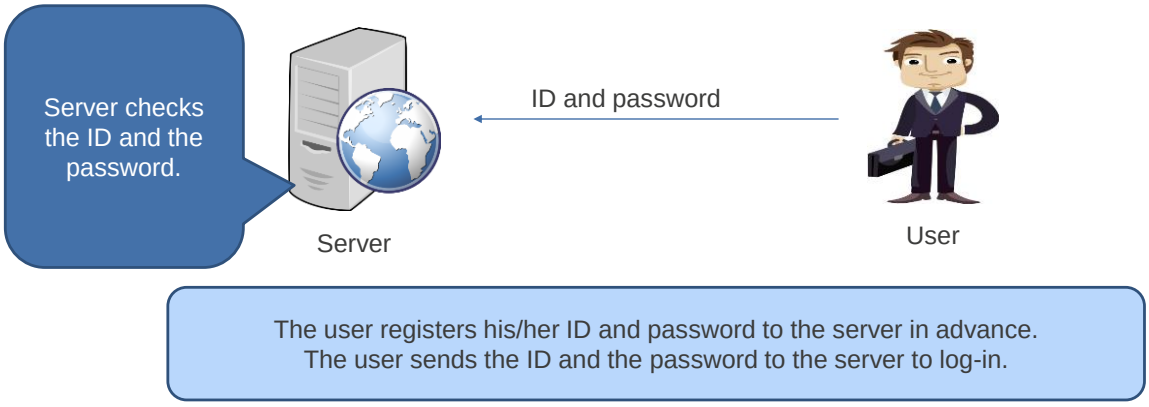


The server must check whether the user has an access right or not.

© 2023 Renesas Electronics Corporation. All rights reserved.

The purpose of an authentication protocol is to identify that the communication peer is the intended person. For example, it is used for authentication when a user tries to log in to a server with his account over the internet.

EXAMPLES OF AUTHENTICATION PROTOCOL: PASSWORDS AUTHENTICATION

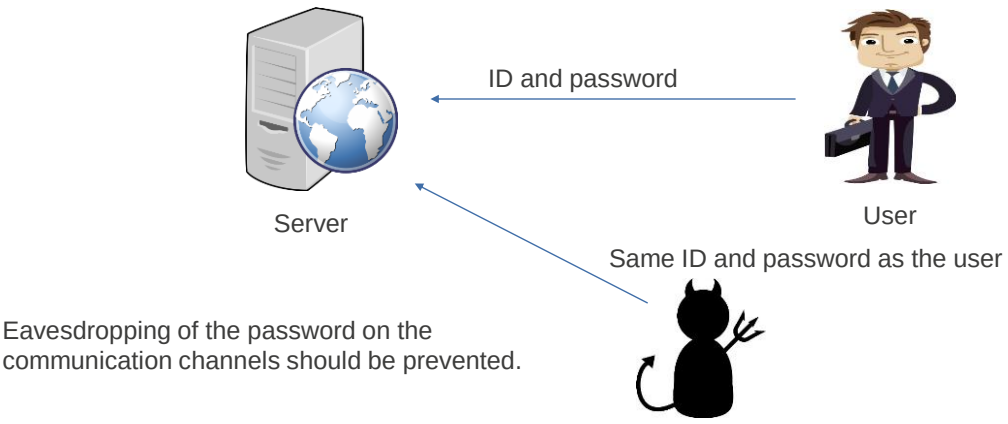


© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL

A commonly used method is password authentication. This is the method where the user registers an ID and a secret password on the server in advance, and then whether sent these data at login matches the data that the server has.

PROBLEMS WITH AUTHENTICATION PROTOCOL: PASSWORDS AUTHENTICATION

- A password might be guessed because a user may set a simple password to be easy to remember.
- Eavesdropping of a password



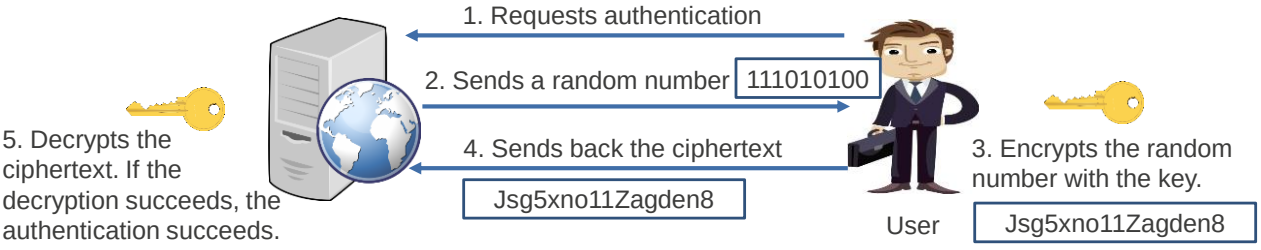
Eavesdropping of the password on the communication channels should be prevented.

© 2023 Renesas Electronics Corporation. All rights reserved.

The password authentication has a problem where an attacker can pass the authentication using illegally obtained ID and password by eavesdropping. Therefore, countermeasures such as encryption are required for communication channels used to send passwords.

AUTHENTICATION PROTOCOL: CHALLENGE & RESPONSE AUTHENTICATION

[Preparation] A user registers his/her cryptographic key with the server in advance.



Points:
- ✓ Only the authorized user can encrypt the random number with the shared key. If the server receives the correct ciphertext, the authentication process succeeds.
- ✓ This protocol prevents replay attacks by using the random number as a challenge.

© 2023 Renesas Electronics Corporation. All rights reserved. RENESAS CONFIDENTIAL

As a method to solve the problem of the password authentication, Challenge & Response authentication exists. In preparation for this method, a user registers his cryptographic key with the server in advance. When logging in, the server sends a random number called a 'challenge' to the user. The user encrypts the challenge with the cryptographic key and returns its encrypted data as a response. The server decrypts the response data using the cryptographic key. If the decrypted data is the same as the original challenge data, the server authenticates the user as the authorized peer. The Challenge & Response authentication is the method where the user sends a ciphertext of a challenge, which is a random number. Therefore, even if a response used by the user in the past is re-sent, an authentication will fail because the decrypted data and the challenge stored on the server are not the same.

Renesas.com

© 2023 Renesas Electronics Corporation. All rights reserved.
RENESAS CONFIDENTIAL